Business Processes Modeling

[Scuola di ballo project n.56 report]

Authors: *Davide Ricci, Guido Trentacapilli*

Professor: *Roberto Bruni*
Master's degree: Data Science & Business Informatics

Indice

Capitolo 1

Introduzione

Il seguente elaborato descrive la modellazione del processo di business ”Scuola di ballo” in cui bisogna gestire le richieste degli allievi di intraprendere i potenziali corsi resi disponibili dalla scuola. Una volta assegnato un corso, l'allievo verrà seguito da un determinato maestro. I principali attori che sono stati identificati per gestire l'intero processo sono:

- **Allievo**
- **Scuola di ballo**
- **Maestro**

In prima istanza, è stata utilizzata la notazione grafica **BPMN** per descrivere la dinamica dell'intero processo suddiviso in diversi pool, uno per ogni attore. La comunicazione è stata realizzata tramite appositi message flows; mentre per rappresentare punti di decisione o diramazione del flusso del processo sono stati utilizzati opportuni **XOR Gateway** e **Event-Based Gateway**. I diagrammi sono stati realizzati attraverso il tool online **BPMN.io**. Nella seconda parte progettuale sono state realizzate e analizzate le **Petri Nets** relative utilizzando **WoPeD** e **Woflan** in modo da verificare le proprietà strutturali del processo ed effettuare un'analisi qualitativa di esso.

Capitolo 2

Diagramma BPMN

2.1 Pool Allievo

Il Pool dell'allievo, dopo lo stato iniziale, inizia il processo inviando una richiesta dei corsi da ballo disponibili, al pool della scuola rimanendo in attesa degli stessi. Una volta ricevuto l'apposito messaggio si sblocca dallo stato di attesa e compie il task di selezione del corso preferito notificando nuovamente la scuola e si mette nuovamente in attesa della ricezione della proposta del primo appuntamento da parte del maestro. A questo punto l'allievo avrà quindi due opzioni, modellate da uno **XOR split**: accettare la proposta oppure formulare una controproposta sempre al maestro. Per gestire questa seconda opzione è stato definito un **Event Based Gateway** in cui l'allievo rimane in attesa di conferma/non conferma della controproposta da parte del maestro. Nel caso di un ulteriore declinazione sarà necessario ripetere il ciclo grazie ad un apposito **XOR join** inserito prima dell'evento di attesa ricezione proposta. Come vedremo nel terzo pool, sia l'allievo che il maestro possono inviare controproposte per pianificare gli appuntamenti ma entrambi i processi possono proseguire solamente in caso di accettazione. Lo step successivo per l'allievo è quello di rimanere in attesa della ricezione di un video preparazione relativo all'appuntamento fissato, una volta ricevuto inizierà un apposito task di analisi del video stesso per poi notificare nuovamente il maestro della visione mettendosi in attesa della ricezione di una serie di prove da svolgere. A questo punto comincerà un task di esecuzione esercizi che dovranno essere giudicati da parte del maestro. E' stato definito nel pool un ulteriore **Event Based Gateway** per modellare l'esito dell'esecuzione di tutte le prove: se l'esito di una prova è negativo sarà necessario ripetere l'esecuzione tramite uno **XOR join** che permette di tornare indietro ed eseguire nuovamente l'esercizio; se l'esito è invece positivo si entra in un ultimo **Event Based Gateway** che modella la decisione da parte del maestro di terminare l'appuntamento oppure di rimanere in attesa della ricezione di nuove prove da eseguire e successivamente inviare. Ad ogni fine appuntamento l'allievo deve eseguire il task di pagamento alla scuola, quindi d'ora in avanti la comunicazione con il maestro viene interrotta. In particolare, tramite uno **XOR split**, l'attore può scegliere di prenotare un nuovo appuntamento e tornare ad inizio processo nella fase di proposta ed eventuale controproposta con il maestro per il successivo appuntamento, oppure può decidere di terminare il corso corrente. quest'ultima opzione mette a disposizione due ulteriori scelte per l'allievo, modellate smepre con uno **XOR split**: selezionare un nuovo corso oppure chiudere definitivamente i rapporti con la scuola, giungendo nello stato finale.

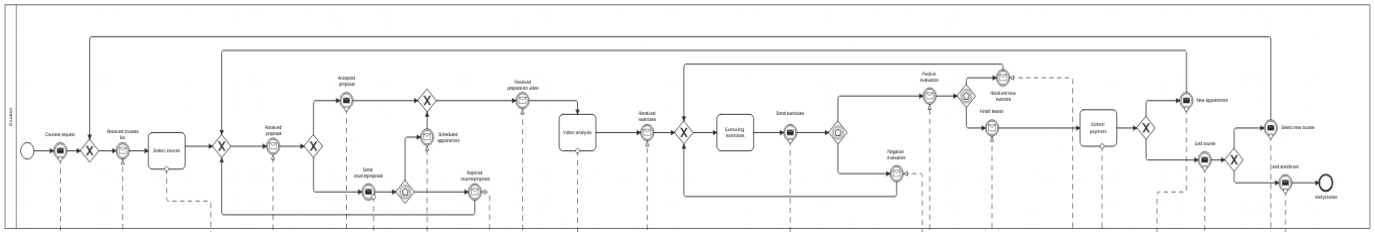


Figura 2.1: Pool allievo

2.2 Pool Scuola di ballo

Il business process della scuola si attiva quando l'allievo effettua la richiesta dei corsi attualmente disponibili. Immediatamente viene notificato il maestro della presenza di un allievo interessato ad iscriversi ad uno dei suoi corsi. Tornando alla scuola, viene effettuato un check dei corsi disponibili attraverso l'accesso ad un **Data Store** ed inviata la lista al richiedente, mettendosi poi in attesa di ricevere il corso selezionato per poi notificare immediatamente il maestro. A questo punto la scuola rimane in attesa della ricezione di pagamento al termine di un appuntamento, seguita da un **Event Based Gateway** che modella la possibilità di richiesta di un nuovo appuntamento da parte dell'allievo oppure la terminazione del corso corrente. Se l'allievo richiede un nuovo appuntamento, verrà messo in contatto nuovamente con il maestro che proporrà nuova data e luogo; altrimenti l'allievo potrà scegliere di terminare il corso. A questo punto è stato definito un ulteriore **Event Based Gateway** per gestire due possibilità: richiesta da parte dello studente di intraprendere un nuovo corso, oppure richiesta di terminare definitivamente i rapporti con la scuola giungendo nello stato finale. Per la prima opzione, tramite uno **XOR join** la scuola dovrà consultare nuovamente i corsi disponibili e reinviarli all'allievo.

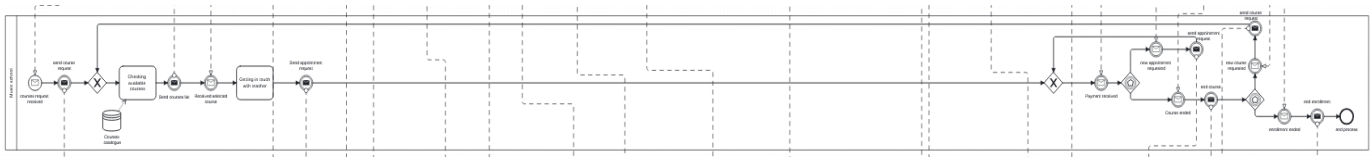


Figura 2.2: Pool scuola di ballo

2.3 Pool Maestro

Il terzo ed ultimo pool descrive il business process del maestro e si avvia una volta ricevuta la notifica, da parte della scuola, di un nuovo studente interessato ai corsi. Il maestro riceve,dalla scuola, la notifica del corso di interesse scelto dall'allievo e invia, sempre allo studente, una proposta di appuntamento per la prima lezione. Qui si trova un **Event Based Gateway** che, a seconda della risposta dello studente, permette al maestro di fissare la lezione oppure ricevere una contro-proposta per l'appuntamento. Per modellare questa situazione è stato definito uno **XOR split**, che permette al maestro di accettare la contro-proposta dello studente, oppure di rifiutare e tornare indietro con una **XOR join**, così da elaborare una nuova proposta da inviare. Una volta fissato l'appuntamento, ovvero superata la **XOR join** che unisce i casi "proposta accettata" e "contro-proposta accettata", il maestro può inviare il video di preparazione allo studente. Ciò viene eseguito tramite un task che comprende l'accesso ad un apposito Data Store, contenente una raccolta di video. Una volta ricevuta la conferma di visione del video, il maestro esegue il task di "preparazione prova" per lo studente e aspetta che venga eseguita. Svolta la prova, la valuta attraverso un task e il processo entra in uno **XOR split** per comunicare allo studente l'esito della prova. In caso di fallimento, si torna indietro con una **XOR join** che mette in attesa il maestro di ricevere nuovamente il risultato. Se la prova è eseguita correttamente, tramite una **XOR split**, il maestro sceglie se mandare una nuova prova, e quindi tornare alla **XOR join**, o terminare la lezione comunicandolo allo studente. Qui, tramite un **Event Based Gateway**, il maestro attende che la scuola gli comunichi la richiesta di un nuovo appuntamento, tornando indietro alla **XOR join** per fissarlo, oppure terminare il corso. In questo ultimo scenario è stato definito un ultimo **Event Based Gateway** che modella il caso in cui lo studente scelga di iniziare un nuovo corso, con il maestro che si mette in attesa di ricevere la nuova comunicazione dalla scuola, e il caso in cui lo studente decide di terminare definitivamente il rapporto.

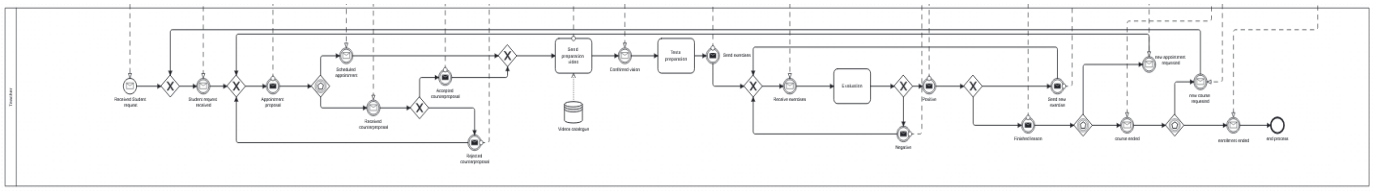


Figura 2.3: Pool maestro

2.4 Diagramma completo

In questa sezione è possibile vedere l'istantanea della **BPMN** completa relativa al nostro caso di studio e, in particolare, come i 3 pool comunicano mediante opportuni message flows.

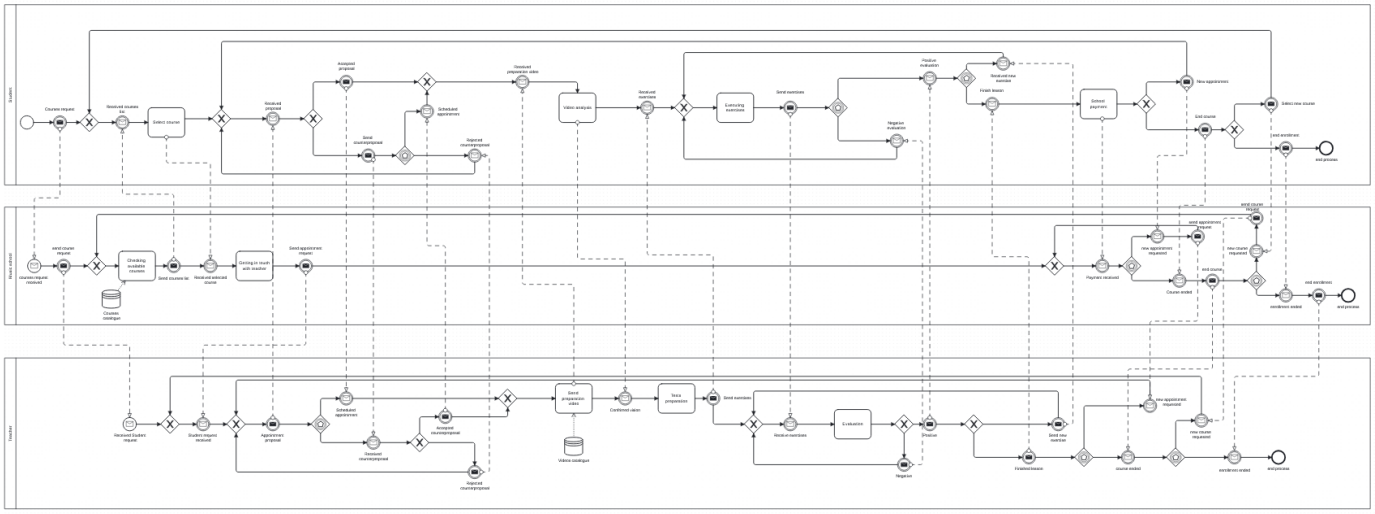


Figura 2.4: Diagramma completo

Capitolo 3

Petri Nets e analisi

In questa seconda fase i processi **BPMN** sono stati tradotti utilizzando l'applicativo **WoPeD** in **Petri Nets** che, come vedremo nelle prossime immagini, sono delle **Workflow net**. **WoPeD** è stato utilizzato anche per l'analisi semantica delle singole workflow net e per la costruzione dei coverability graph, mentre per il sistema di workflow net completo è stato utilizzato **Woflan**, convertendo opportunamente la rete in formato "tpn". Nel seguito vengono descritti i passaggi della trasformazione del diagramma **BPMN**:

- Sequence/Message flows tradotti in places
- Events/Activities/Exclusive Gateways tradotti in transitions
- Event Based Gateways sostituiti da un'unica piazza di input contenente nel postset una transizione per ogni potenziale alternativa.
- Aggiunta di un initial e final place per ogni pool
- In particolare ogni Gateway è stato desugarized

3.1 Allievo

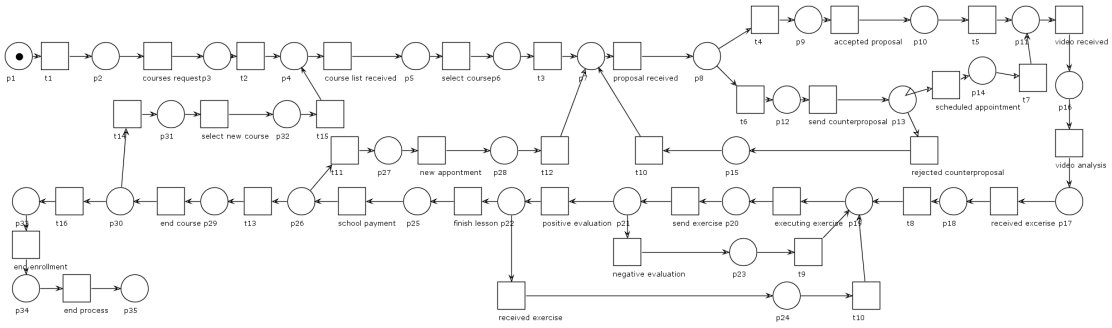


Figura 3.1: Workflow Net Allievo

La rete dello studente risulta avere **35 places, 40 transitions e 80 arcs**. Dall'analisi effettuata risulta che la rete è effettivamente una workflow net in quanto presenta:

- Unico place iniziale che non presente archi in ingresso
- Unico place finale che non presenta archi in uscita
- ogni place e transition appartengono ad almeno un cammino dal place iniziale a quello finale

Avendo un unico **S-Component** ed essendo i preset e postset delle transizioni composti da un unico place la rete è una **S-net**. Di conseguenza, essendo presente un unico token, si può concludere che la rete è anche **sound**, **safe** e **free-choice** quindi **live** e **bounded**. Inoltre dalla analisi semantica si può dedurre che la rete è anche **S-Coverable** in quanto presenta un unico **S-Component** che copre tutta la rete ed è **Strongly-connected** perchè tutti i nodi presentano archi in entrata, quindi possono essere raggiunti. Infine sono preservate anche le proprietà di **deadlock-free**, **liveness** e **well-structured** in quanto non sono presenti **PT-Handles**. La rete non è però una **T-Net** in quanto alcuni places presentano 2+ input/output transitions.

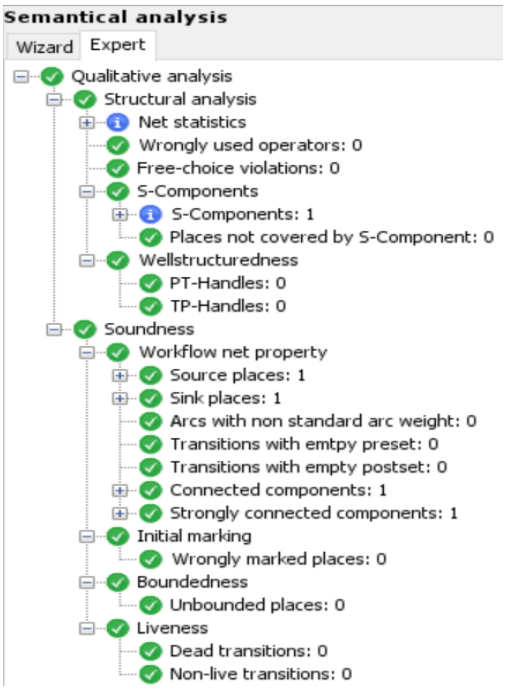


Figura 3.2: Analisi semantica Allievo

Per quanto riguarda il Coverability Graph relativo, si può notare che è composto da **35 vertici e 40 archi**.

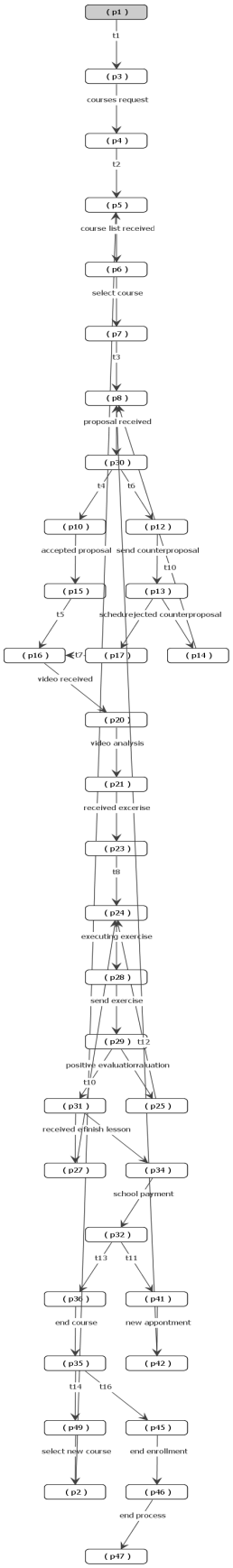


Figura 3.3: Coverability Graph Allievo

3.2 Scuola di ballo

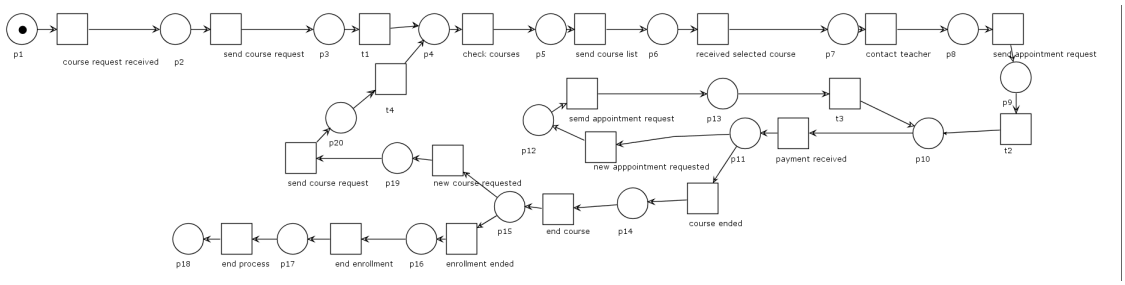


Figura 3.4: Workflow Net Scuola

Per quanto riguarda la net della scuola di ballo, essa presenta **20 places, 21 transitions e 42 arcs**. L'analisi semantica effettuata è analoga al caso precedente e, le proprietà verificate e non, risultano essere le stesse.

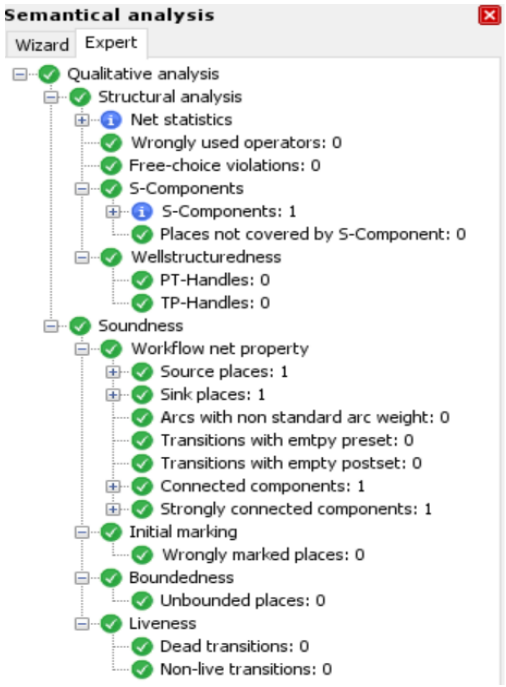


Figura 3.5: Analisi semantica Scuola

Il Coverability Graph di questo attore presenta **20 vertici e 31 archi** e, come per il precedente, è possibile verificare che non esistono **dead task**.

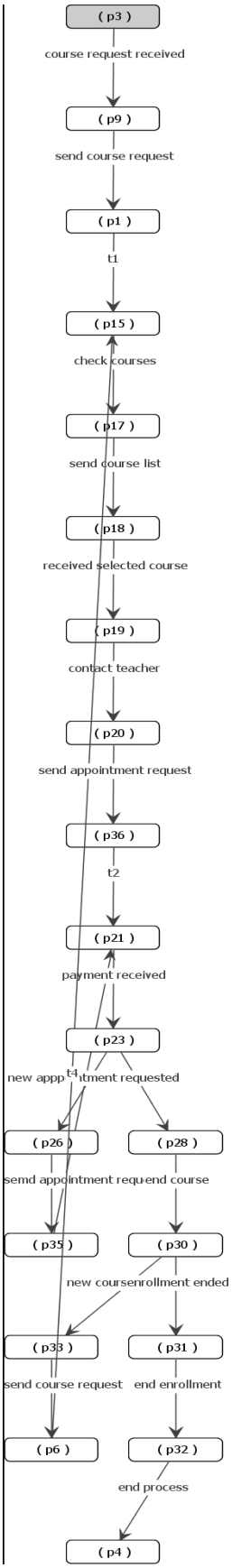


Figura 3.6: Coverability Graph Scuola

3.3 Maestro

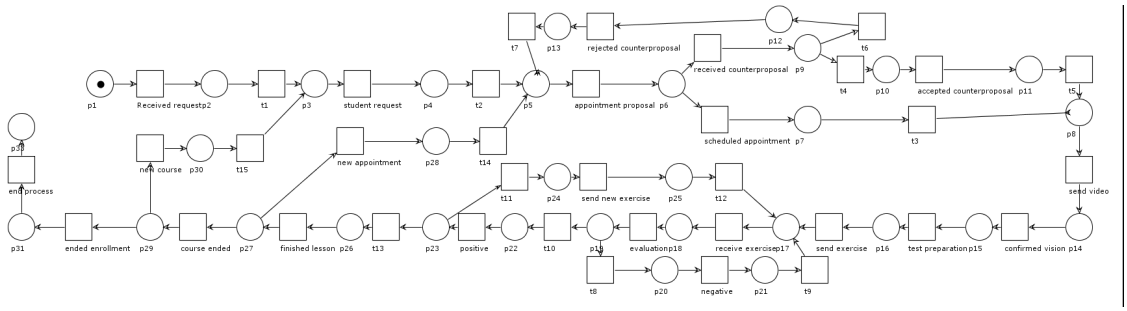


Figura 3.7: Workflow Net Maestro

Infine, la petri net risultante relativa al maestro presenta **32 places, 37 transitions e 74 arcs**. Seguendo sempre la solita metodologia di analisi tramite **WoPeD** si arriva alle solite conclusioni strutturali e semantiche: la rete risulta essere una **workflow net, free-choice, S-net** dunque **bounded,sound e safe**. Inoltre sono sempre preservate le proprietà di **deadlock-free, strongly connectedness e well structuredness**. Anche in questo caso non risulta però essere una **T-Net**.

Figura 3.8: Analisi semantica Maestro

Nel seguito il Coverability Graph relativo.

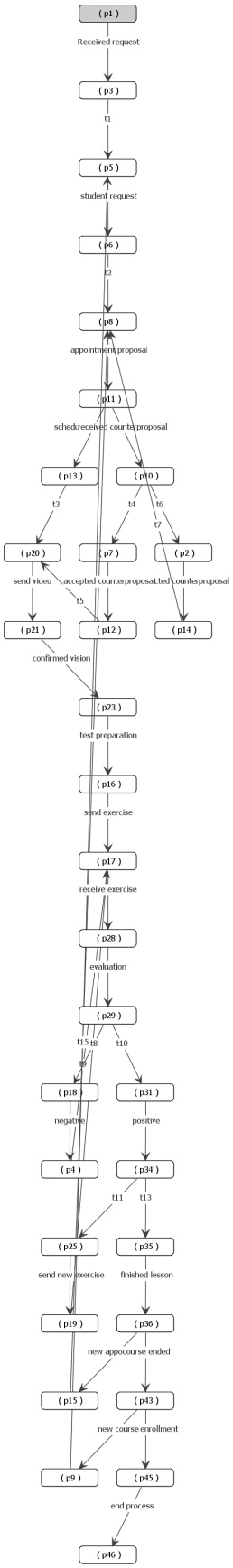


Figura 3.9: Coverability Graph Maestro

3.4 Workflow net completo

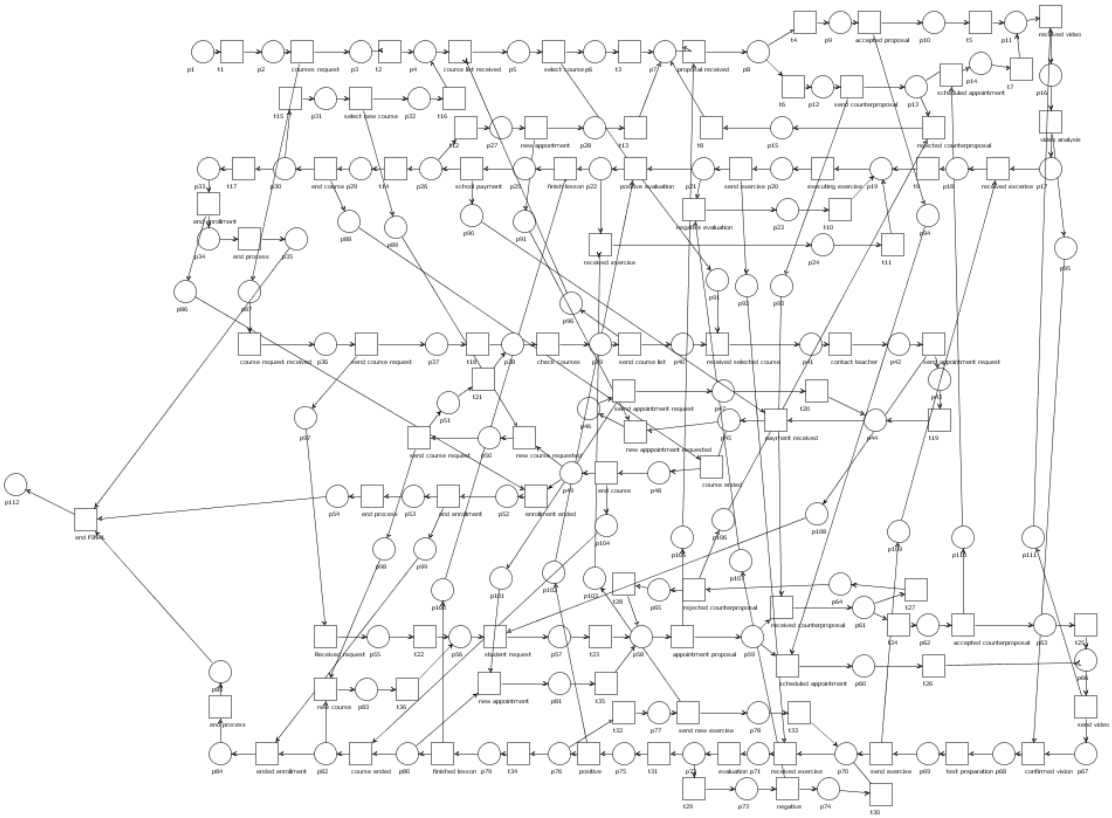


Figura 3.10: Workflow Net completa

In ultima istanza i vari workflow net sono stati trasformati in moduli di workflow, realizzando un sistema completo mediante l'introduzione di un unico **initial place** e un unico **final place**. Per garantirne l'unicità è stato considerato come place di avvio quello della Petri Net dell'allievo mentre per lo stato finale è stato definito un end place raggiungibile tramite una **AND transition** che riceve in input gli stati finali dei singoli 3 workflow relativi ai 3 attori. La rete completa è quindi composta da un unico initial place contenente un token che abilita la richiesta dell'allievo alla scuola e da un unico end place che codifica la terminazione del rapporto tra allievo-scuola-maestro. Per quanto riguarda l'analisi semantica e strutturale è stato utilizzato l'applicativo **Woflan** a causa di svariati problemi di freezing riscontrati su **WoPeD** data la natura complessa della workflow net completa. In definitiva la rete finale è composta da **113 places e 99 transitions** e risulta preservare le proprietà di **Soundness (boundedness, liveness)**. A differenza dei singoli moduli, la rete non è **Free-Choice** nè una **S-Net** in quanto contiene una violazione causata dai vari message-flows corrispondenti ad ogni **Event Based Gateway**. Infine è risultato che non presenta nemmeno le proprietà di **Well Structuredness** in quanto presenta **364 PT-HANDLES e 250 TP-Handles**. In figura 3.12 è raffigurato il complesso **coverability graph** ma per un'analisi più attenta è consigliata la generazione real time tramite **WoPeD**.

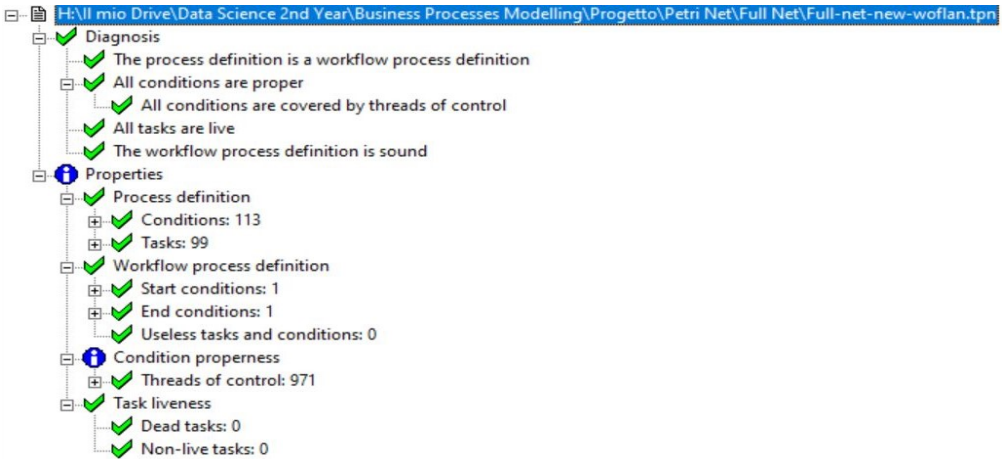
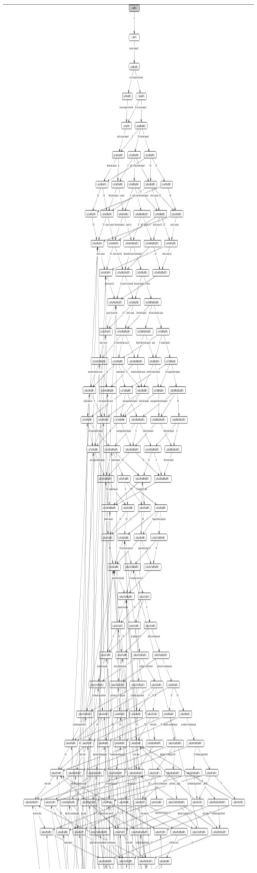


Figura 3.11: Analisi semantica rete completa



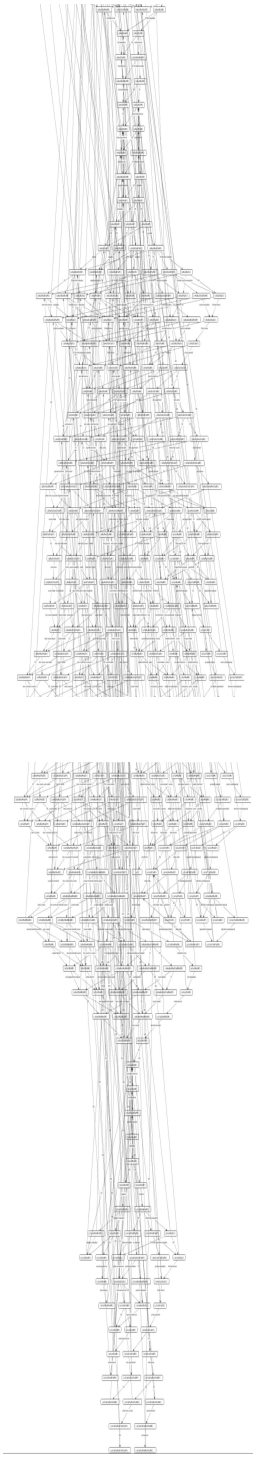


Figura 3.12: Coverability Graph